



# Artificial Software Diversification for WebAssembly

JAVIER CABRERA-ARTEAGA

Doctoral Thesis  
Supervised by  
Benoit Baudry and Martin Monperrus  
Stockholm, Sweden, 2023

TRITA-EECS-AVL-2020:4  
ISBN 100-

KTH Royal Institute of Technology  
School of Electrical Engineering and Computer Science  
Division of Software and Computer Systems  
SE-10044 Stockholm  
Sweden

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges  
till offentlig granskning för avläggande av Teknologie doktorexamen i elektroteknik  
i .

© Javier Cabrera-Arteaga , date

Tryck: Universitetsservice US AB

**Abstract**

[1]

**Keywords:** Lorem, Ipsum, Dolor, Sit, Amet

**Sammanfattning**

[1]

## LIST OF PAPERS

1. ***WebAssembly Diversification for Malware Evasion***  
**Javier Cabrera-Arteaga**, Tim Toady, Martin Monperrus, Benoit Baudry  
*Computers & Security, Volume 131, 2023*  
<https://www.sciencedirect.com/science/article/pii/S016740482302067>
2. ***Wasm-mutate: Fast and Effective Binary Diversification for WebAssembly***  
**Javier Cabrera-Arteaga**, Nicholas Fitzgerald, Martin Monperrus, Benoit Baudry
3. ***Multi-Variant Execution at the Edge***  
**Javier Cabrera-Arteaga**, Pierre Laperdrix, Martin Monperrus, Benoit Baudry  
*Conference on Computer and Communications Security (CCS 2022), Moving Target Defense (MTD)*  
<https://dl.acm.org/doi/abs/10.1145/3560828.3564007>
4. ***CROW: Code Diversification for WebAssembly***  
**Javier Cabrera-Arteaga**, Orestis Floros, Oscar Vera-Pérez, Benoit Baudry, Martin Monperrus  
*Network and Distributed System Security Symposium (NDSS 2021), MADWeb*  
<https://doi.org/10.14722/madweb.2021.23004>
5. ***Superoptimization of WebAssembly Bytecode***  
**Javier Cabrera-Arteaga**, Shrinish Donde, Jian Gu, Orestis Floros, Lucas Satabin, Benoit Baudry, Martin Monperrus  
*Conference Companion of the 4th International Conference on Art, Science, and Engineering of Programming (Programming 2021), MoreVMs*  
<https://doi.org/10.1145/3397537.3397567>
6. ***Scalable Comparison of JavaScript V8 Bytecode Traces***  
**Javier Cabrera-Arteaga**, Martin Monperrus, Benoit Baudry  
*11th ACM SIGPLAN International Workshop on Virtual Machines and Intermediate Languages (SPLASH 2019)*  
<https://doi.org/10.1145/3358504.3361228>



## ACKNOWLEDGEMENT



## ACRONYMS

List of commonly used acronyms:

**Wasm** WebAssembly



# Contents

<b>List of Papers</b>	<b>iii</b>
<b>Acknowledgement</b>	<b>v</b>
<b>Acronyms</b>	<b>vii</b>
<b>Contents</b>	<b>1</b>
<b>I Thesis</b>	<b>3</b>
<b>1 Introduction</b>	<b>5</b>
1.1 Background . . . . .	5
1.2 Problem statement . . . . .	5
1.3 Automatic Software diversification requirements . . . . .	5
1.4 List of contributions . . . . .	5
1.5 Summary of research papers . . . . .	6
1.6 Thesis outline . . . . .	6
<b>2 Background and state of the art</b>	<b>7</b>
2.1 WebAssembly . . . . .	7
2.1.2 WebAssembly ecosystems . . . . .	7
2.1.3 Binary format . . . . .	7
2.1.4 Types . . . . .	9
2.1.5 Memory model . . . . .	9
2.1.6 Execution model . . . . .	9
2.2 Software diversification . . . . .	9
2.2.2 Generating Software Diversification . . . . .	9
2.2.3 Variants generation . . . . .	9
2.2.4 Variants equivalence . . . . .	9

2.3	Exploiting Software Diversification . . . . .	9
2.3.2	Defensive Diversification . . . . .	9
2.3.3	Offensive Diversification . . . . .	9
<b>3</b>	<b>Automatic Software Diversification for WebAssembly</b>	<b>11</b>
3.1	CROW: Code Randomization of WebAssembly . . . . .	12
3.1.2	Variants' generation . . . . .	12
3.1.3	Constant inferring . . . . .	14
3.1.4	Combining replacements . . . . .	14
3.1.5	CROW instantiation . . . . .	15
3.2	MEWE: Multi-variant Execution for WebAssembly . . . . .	17
3.2.2	Multivariant generation . . . . .	17
3.3	WASM-MUTATE: Fast and Effective Binary for WebAssembly . . . . .	20
3.3.2	WebAssembly Rewriting Rules . . . . .	21
3.3.3	Extending peephole meta-rules with custom operators . . . . .	25
3.3.4	E-graphs . . . . .	27
3.3.5	Random e-graph traversal for variants generation . . . . .	27
3.3.6	WASM-MUTATE instantiation . . . . .	29
3.4	Comparing CROW, MEWE, and WASM-MUTATE . . . . .	31
3.4.2	Technology and approach . . . . .	31
3.4.3	Strength of the generated variants . . . . .	31
3.4.4	Security guarantees . . . . .	32
3.5	Conclusions . . . . .	34
<b>4</b>	<b>Exploiting Software Diversification for WebAssembly</b>	<b>35</b>
4.1	Offensive Software Diversification . . . . .	35
4.1.2	<b>Use case 1:</b> Automatic testing and fuzzing of WebAssembly consumers . . . . .	35
4.1.3	<b>Use case 2:</b> WebAssembly malware evasion . . . . .	35
4.2	Defensive Software Diversification . . . . .	35
4.2.2	<b>Use case 3:</b> Multivariant execution at the Edge . . . . .	35
4.2.3	<b>Use case 4:</b> Speculative Side-channel protection . . . . .	35
<b>5</b>	<b>Conclusions and Future Work</b>	<b>37</b>
5.1	Summary of technical contributions . . . . .	37
5.2	Summary of empirical findings . . . . .	37
5.3	Summary of empirical findings . . . . .	37
5.4	Future Work . . . . .	37

CONTENTS	3
----------	---

<b>II Included papers</b>	<b>39</b>
Superoptimization of WebAssembly Bytecode	<b>43</b>
CROW: Code Diversification for WebAssembly	<b>45</b>
Multi-Variant Execution at the Edge	<b>47</b>
WebAssembly Diversification for Malware Evasion	<b>49</b>
Wasm-mutate: Fast and Effective Binary Diversification for WebAssembly	<b>51</b>
Scalable Comparison of JavaScript V8 Bytecode Traces	<b>53</b>



## **Part I**

# **Thesis**

